Université Paris Diderot Master 1

Circuits et architecture

TD 9 (TP 3) : utilisation du mini-processeur LC3

Mode d'emploi

Pour utiliser ce microprocesseur :

- écrire un programme et le mettre dans un fichier mémoire ou utiliser un des fichiers fournis
- lancer la simulation
- charger le programme dans la RAM (avant dernier icône en haut à droite)
- basculer l'interrupteur "Reset" (à gauche) sur On
- appuyer sur Tab jusqu'à ce que que le registre IR libère la valeur de la première instruction (le LED qui est au dessus devient rouge et indique sa valeur)
- basculer l'interrupteur "Reset" (à gauche) sur Off
- appuyer sur Tab et observer le déroulement du programme

Pour définir le contenu de la mémoire, vous devez écrire un fichier `.mem` qui commence par `memory Memoire.m` (c'est le chemin vers le module), puis un offset terminé par un slash et enfin une suite de valeurs hexadécimales (voir le fichier de l'exercice 2).

Rappel du jeu d'instruction

Comme annoncé au TD7, seules les instructions `ADD, AND, BR, JMP, LDR, NOT, STR` sont "câblées". Nous avons utilisé une méthode assez différente de celle présentée en cours, afin de simplifier le circuit. En particulier il n'y a pas de bus unique, pas de registre d'adresse mémoire. Toutes les instructions sont exécutées en 2 cycles d'horloge : pendant le premier cycle, l'instruction est lue en mémoire et stockée dans le registre d'instruction (en bas à gauche), pendant le deuxième cycle, elle est exécutée et le compteur de programme est remis à jour (incrémenté ou autre).

Exercices

Exercice 1 : Si vous ne l'avez pas fait dans le TP précédent, téléchargez le circuit `LC-3-horloge.v`, posez des sondes (double-clic) sur l'horloge, en entrée du diviseur de fréquences et sur ses deux sorties notées "etape1"

et “etape2”. Basculez l'interrupteur “Reset” sur On, appuyez sur Tab pour faire avancer le temps au moins de 2 cycles d'horloge, basculez l'interrupteur “Reset” sur Off et refaites avancer le temps. Observez le chronogramme.

Exercice 2 : Charger maintenant le circuit `LC-3.v`, puis recopiez sur votre répertoire le programme `AddSimple.mem` à l'adresse `http://www.liafa.jussieu.fr/~amicheli/Ens/Archi/`. Ce programme correspond au code assembleur suivant :

```
adresse    instruction                              code hexa

x0000      AND R0,R0,0                              5020
x0001      ADD R0,R0,5  ; R0 <- 5                   1025
x0002      AND R1,R1,0                              5260
x0003      ADD R1,R1,6  ; R1 <- 6                   1266
x0004      ADD R2,R1,R0 ; R2 <- R1 + R0             1440
x0005      AND R0,R0,0  ; R0 <-0                    5020
x0006      STR R2,R0,#16                            7410
```

Faites une simulation complète de ce programme. On peut à tout moment rentrer dans le banc de registre en cliquant dessus pour le sélectionner et en appuyant sur '>' (pour revenir au mini-processeur cliquez sur '<').

Pour vérifier que la mémoire a bien été modifiée, sauvegardez le contenu de la mémoire. Pour cela, rentrez dans le module 'Memoire' (cliquez sur '>'), sélectionnez le composant et cliquez sur la dernière icône en haut à droite.

Exercice 3 :

- Comment se fait le décodage d'une instruction ?
- Comment fait le processeur pour
  - Charger une constante ?
  - Lire ou écrire en mémoire ?
  - Faire une addition ?

Exercice 4 : Ecrire un programme qui calcule les sommes partielles d'une suite de 10 entiers, en écrasant la suite en mémoire. Par exemple (5, 2, 10, 3, 6, 120, 20, 10, 0, 1) est remplacée par (5, 7, 17, 20, 26, 146, 166, 176, 176, 177).

Vous chargerez les 10 entiers en mémoire à la fin du fichier contenant votre programme en écrivant pour chaque entier l'adresse hexadécimale de la mémoire, suivie d'un /, puis du nombre entier à charger à cette adresse.

Faites une simulation de ce programme et sauvegarder le contenu de la mémoire. Assurez-vous que le programme a bien fait ce qu'il fallait.

Exercice 5 : “Overclocker” le processeur en diminuant la période (i.e augmenter la fréquence) de l'horloge. Que se passe-t-il ? À quel seuil ?